Considerations in Choosing an Educational Robot

2016 ITEEA | Washington, DC

J. Mativo, R. Hill, & W. Immonen | University of Georgia

Elementary STEM & robotics

- Context of project is elementary engineering course
- Students are undergraduate majors in early childhood education
- Focusing on integrating hands-on engineering design activities with STEM
- Not an after school activity; ALL students participate
- Not a new subject or content area; teaching the same content with enriched methods

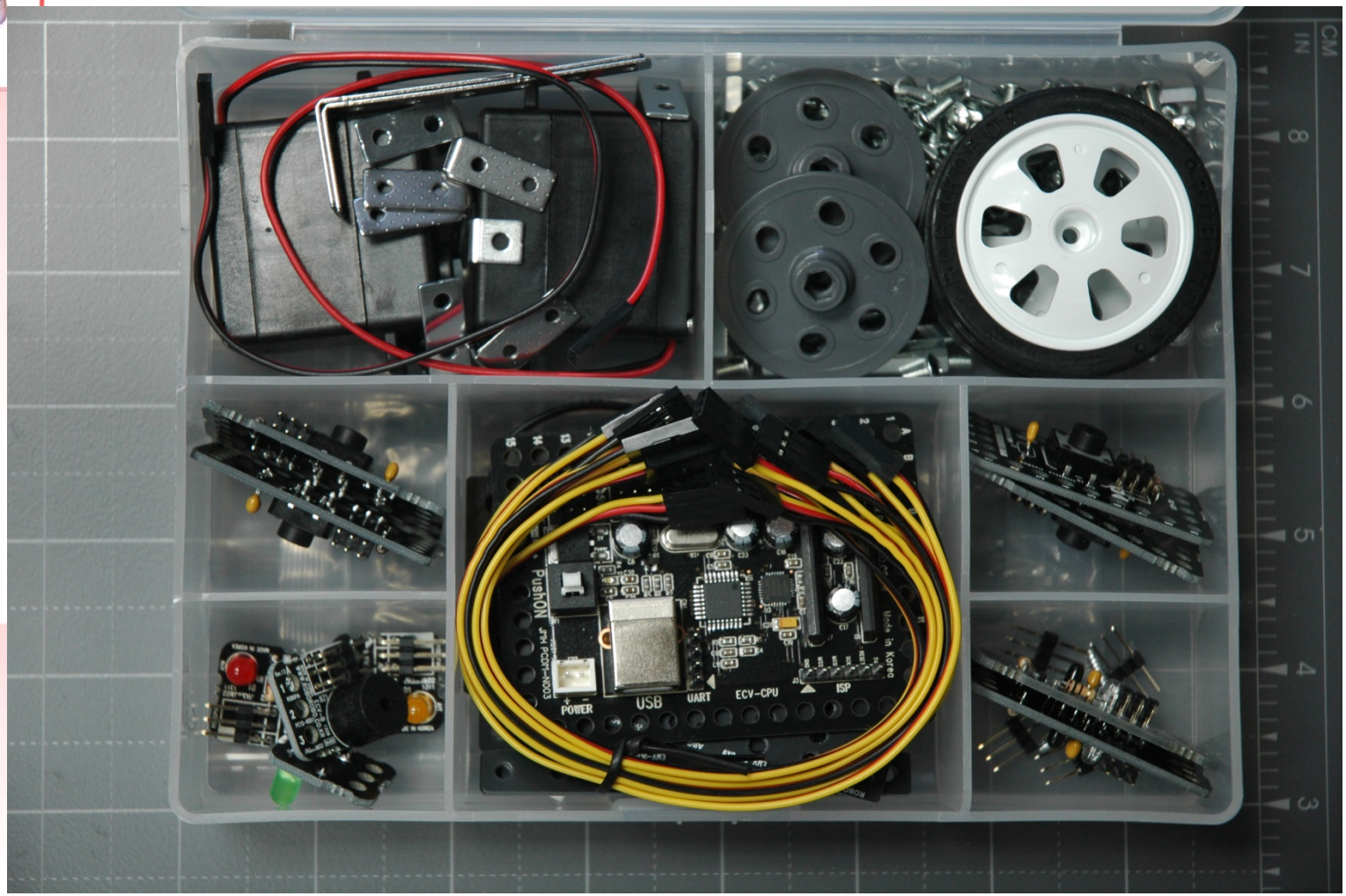
Starting point was NXT

- Lego Mindstorms NXT
- By doing so was targeting 4th and 5th grade students
- Programming difficult for primary grades; UGA students handled it fine
- Troubling to answer when students asked how much it costs
- Lower cost equipment and software needed for integrated STEM robotics in elementary classrooms

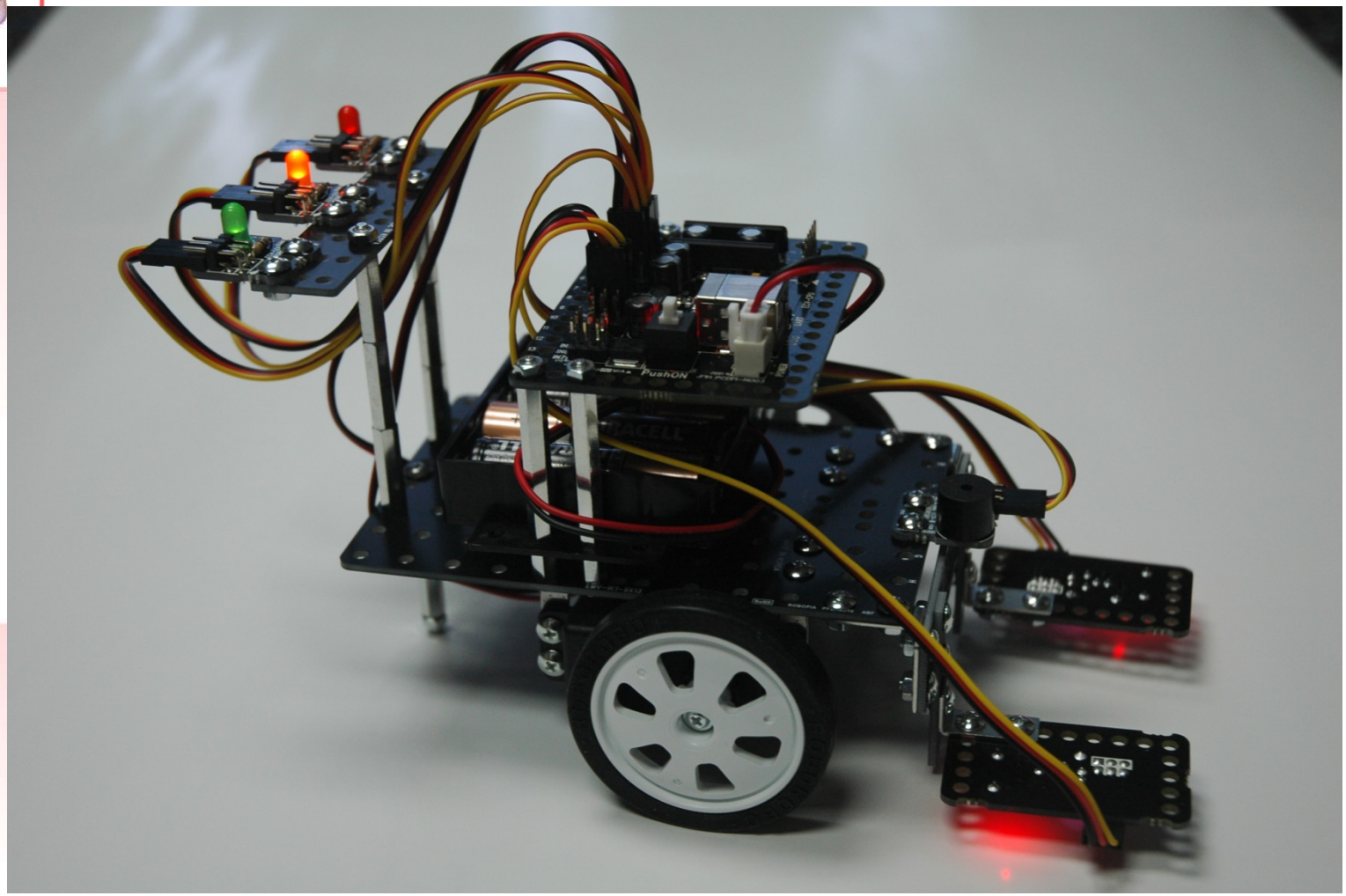
Key characteristics of hardware

- Easy to assemble
- Easy to program
- Does not look like a toy
  - Builds self-efficacy when learners succeed
  - Learners more serious when hardware looks like a real machine
- Economical (educational price < $75) for level 1 kit

Entire level 1 kit in 20cm x 15cm case



IR sensors and LED lights



Simple to use programming

Introduction

- Guiding consideration: R(i)
  - Your purpose determines your robot
- Considerations when selecting an educational robot may include
  - Availability
  - Capability
  - Use-ability
  - Affordability
  - Student level

My Robot Time

- Hardware
  - Microcontroller: Atmega32A, 8 bit CMOS, 32 KB memory, 32 I/O
  - Peripherals: sensors, motors
- Software: MRT
- Additional Information: sold in kits

MRT software

MRT[IR-LR-motors.prj]

File(F) Edit(E) Set(S) Build(B) Help View

NEW OPEN SAVE COM MAKE DOWNLOAD

Inputs

Romocon Touch Mic

IR Cds Custom

Counter

Outputs

LED Custom Servo

Motor Buzzer LCD

Debug

Flow

Stop Repeat Delay

Goto

Program Code

Action	No
Program Start	0
IR : [IN3]=[SENSE]{	1
Motor : [LEFT_MOTOR_1]=[Backward,-10],[RIGHT_MOTOR_1]=	2
End	3
IR : [IN3]=[NONE]{	4
Motor : [LEFT_MOTOR_1]=[Forward,5],[RIGHT_MOTOR_1]=	5
End	6
Program End	7

Output

Connect Clear

J:\course_mat_robotics\mrt\Skier\IR-LR-motors.prj

RoboRobo

- Hardware
  - Microcontroller: Atmega8A, 8 bit, 8 KB flash memory, 16 I/O
  - Peripherals: motors, sensors
- Software: rogic
- Additional Information: sold in kits

Rogic software

Rogic 3.3.0.1

File Setting Mode About

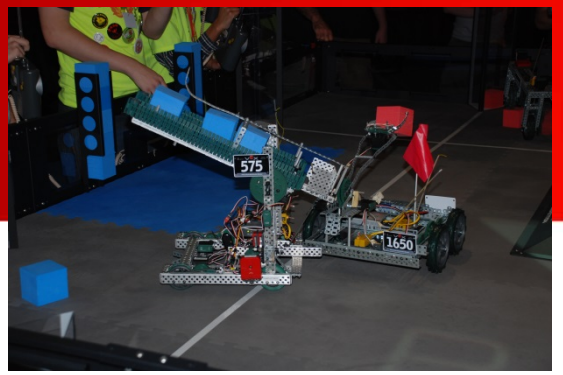
Down load Run Stop WWW Exit

Functions
Chips
Delay
While
IF Else

DC Motor
Servo
Voice
Dof Matrix
ON
OFF
Variable
Calculate

Start
While ∞
IF Port 1
YES NO
On 1
Off 1
Delay .1
Delay .1
End If

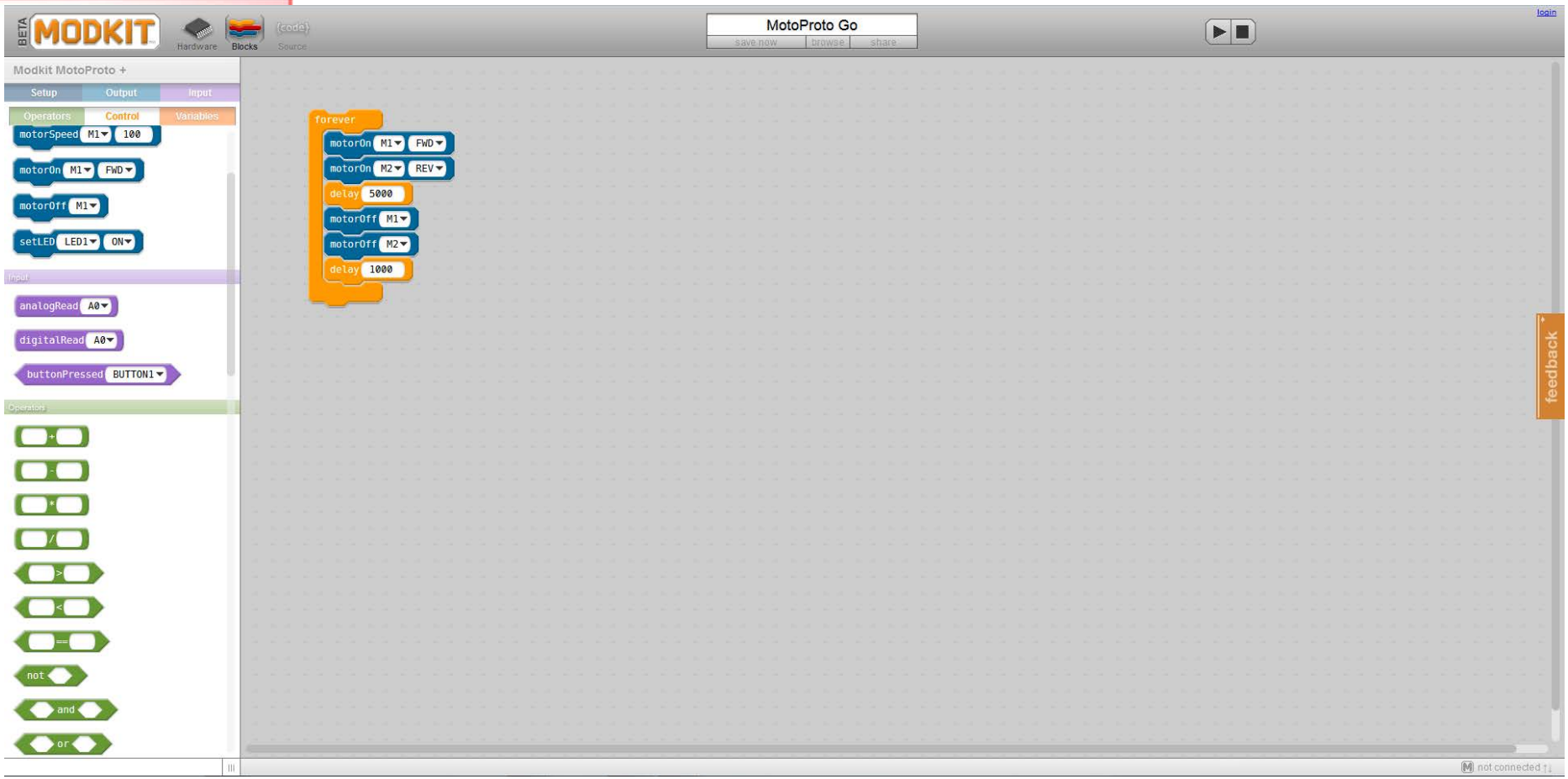
VEX (IQ)



- Hardware
  - Microcontroller: Texas Instruments Tiva ARM Cortex-M4 Processor, 256k flash, 32k ram, 12 I/O
  - Peripherals: motors, sensors(line tracking, ultrasonic range, light, optical shaft, bumper, potentiometer, gyro, accelerometer), LCD display, flashlight, speaker, wireless controller (joysticks)
- Software: ROBOTC, Modkit, C
- Additional information: there are multiple VEX robotics kits to choose from


t
e
a
m
w
o
r
k

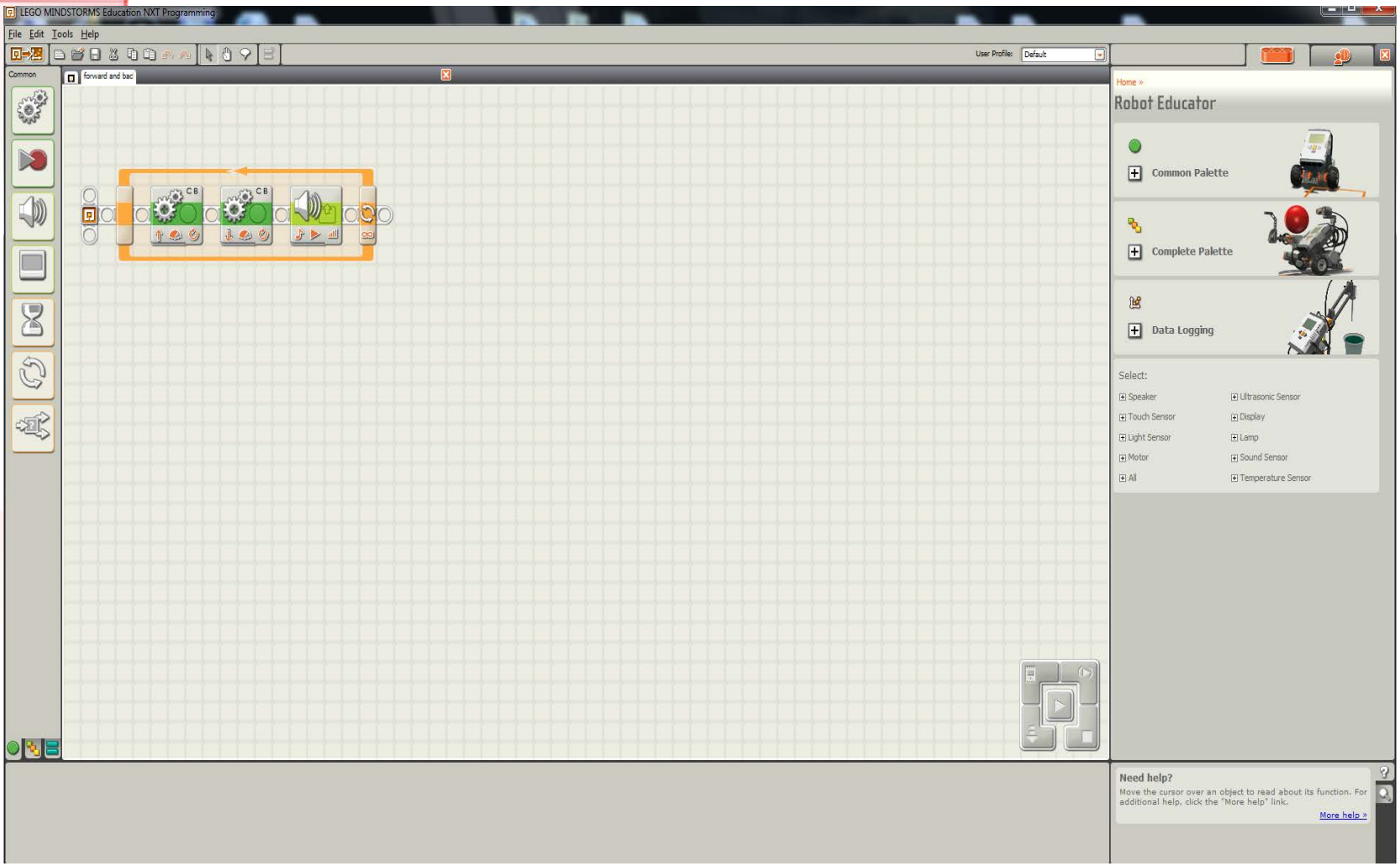
VEX ModKit


BETA
MODKIT
Hardware
Blocks
{code}
Source
MotoProto Go
save now
browse
share
login
Modkit MotoProto +
Setup
Output
Input
Operators
Control
Variables
motorSpeed M1 100
motorOn M1 FWD
motorOff M1
setLED LED1 ON
Input
analogRead A0
digitalRead A0
buttonPressed BUTTON1
Operators
+
-
*
/
>
<
==
not
and
or
forever
motorOn M1 FWD
motorOn M2 REV
delay 5000
motorOff M1
motorOff M2
delay 1000
feedback
not connected

Lego NXT

- Hardware
  - Microcontroller: At91SAM7S256, 32 bit ARM7, 64 KB memory, 32 I/O
  - Peripherals: infrared seeker, gyro, receiver, and motors
- Software: NXT-G programming, ROBOTC, labVIEW, and other possible languages can be used

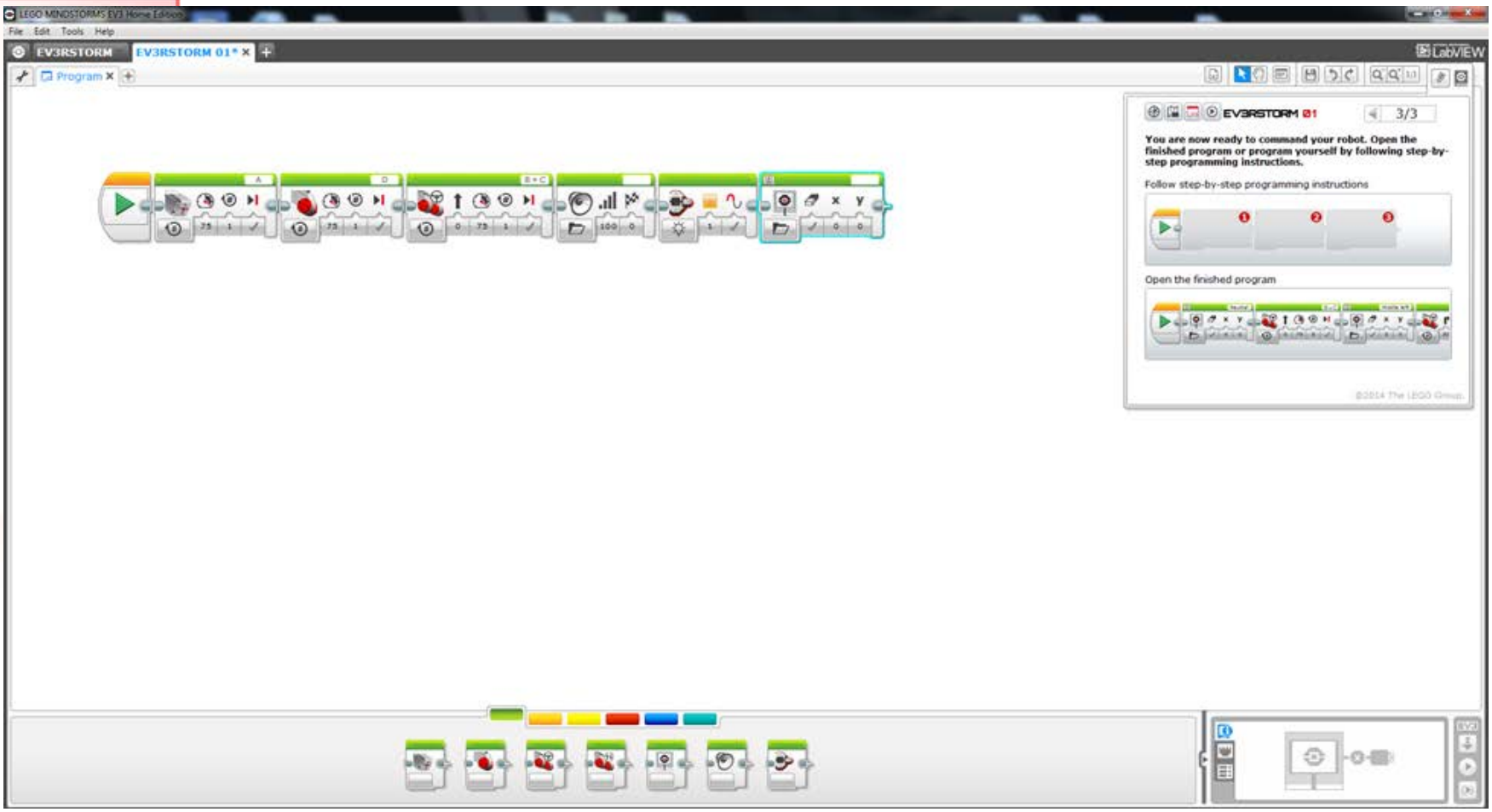
Lego Mindstorms NXT software



Lego EV3

- Hardware
  - Microcontroller: AM1808(ARM9), 32 bit, 64 KB memory, 32 I/O
  - Peripherals: sensors(touch, color, gyro, ultrasonic and infrared), motors, servos
- Software: EV3 programming, ROBOTC, labVIEW, and other possible languages can be used

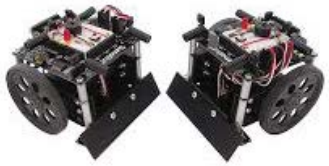
Lego Mindstorms EV3 software



Sumobot

- Hardware
  - Base package: PIC18F4550, 8 bit, 32 KB memory, 35 I/O
  - Peripherals : motors, sensors(rangefinder, ultrasonic, IR, PIR, sound, humidity, gas, gyro), servos, speakers, GPS, SD card adapters, lights, displays (including touchscreen), readers (RFID, smartcard)
- Software: BASIC STAMP

BASIC Stamp



BASIC Stamp - D:\Users\William\Desktop\27402-Applied-Robotics-SumoBot-Code-v1.0\AppliedSumo-Code-v1.0.pdf\SumoWrestler.bs2

File Edit Directive Run Help

D:\...\AppliedSumo-Code-v1.0.pdf

SumoWrestler.bs2

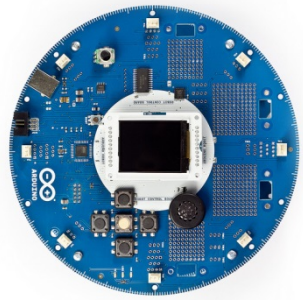
William
.spss
Adobe Flash Builder 4.6
AppData
Application Data
Contacts
Creative Cloud Files
Desktop
27402-Applied-Robotics-SumoBot-C
AppliedSumo-Code-v1.0.pdf
portfolio
Documents
Downloads
Dropbox

CompilerDirectives.bs2
Forward100Pulses.bs2
ForwardLowTimeTest.bs2
FrontAndSideIrNavigation.bs2
FrontIrNavigation.bs2
IrInterferenceSniffer.bs2
LookupExample.bs2
PushbuttonLed.bs2
PushbuttonMode.bs2
QtiPulseDecayTrick.bs2
QtiPulseTrickLeft.bs2
QtiSelfCalibrate.bs2
ResetAndStartMode.bs2
ResetButtonCounter.bs2
SearchPatternAndAvoidTawara.bs2
SensorsByteDisplay.bs2
SensorsWithFlagByte.bs2
SensorsWithTempVariables.bs2
ServoControlExample.bs2
ServoControlWithLookup.bs2
SumoWrestler.bs2
SumoWrestlerWithDataLogMode.bs2
SumoWrestlerWithDebugMode.bs2
SymbolNamesVsAddressContents.bs2
TestAllSensors.bs2
TestFrequencyResponse.bs2
TestFrontIrObjectDetectors.bs2
TestFrontQtiLineSensors.bs2
TestLedSpeaker.bs2
TestPushButton.bs2
TestResetButton.bs2
TestSideIrObjectDetectors.bs2
TestSumoWrestler.bs2
ThreeVariablesManyJobs.bs2

BASIC Stamp files (*.bs1;*.bas;*.bs2;*.bse;*.bsx

```
' -----[ Title ]-----------------------------------------------------------
' Applied Robotics with the SumoBot - SumoWrestler.bs2
' SumoWrestler.bs2 modified so that each state is contained by a
' subroutine.

' {$STAMP BS2}                          ' Target = BASIC Stamp 2
' {$PBASIC 2.5}                         ' Language = PBASIC 2.5

' -----[ I/O Definitions ]-------------------------------------------------

ServoLeft       PIN     13              ' Left servo connected to P13
ServoRight      PIN     12              ' Right servo connected to P12

qtiPwrLeft      PIN     10              ' Left QTI on/off pin P10
qtiSigLeft      PIN     9               ' Left QTI signal pin P9

qtiPwrRight     PIN     7               ' Right QTI on/off pin P7
qtiSigRight     PIN     8               ' Right QTI signal pin P8

DummyPin        PIN     6               ' I/O pin for pulse-decay P6

LedSpeaker      PIN     5               ' LED & speaker connected to P5

IrLedLS         PIN     2               ' Left IR LED connected to P2
IrSenseLS       PIN     1               ' Left IR detector to P1

IrLedLF         PIN     4               ' Left IR LED connected to P4
IrSenseLF       PIN     11              ' Left IR detector to P11

IrLedRF         PIN     15              ' Right IR LED connected to P15
IrSenseRF       PIN     14              ' Right IR detector to P14

IrLedRS         PIN     3               ' Right IR LED connected to P3
IrSenseRS       PIN     0               ' Right IR detector to P0

' -----[ Constants ]-------------------------------------------------------

' SumoBot maneuvers

Forward         CON     0               ' Forward
Backward        CON     1               ' Backward
RotateLeft      CON     2               ' RotateLeft
RotateRight     CON     3               ' RotateRight
PivotLeft       CON     4               ' Pivot to the left
PivotRight      CON     5               ' Pivot to the right
CurveLeft       CON     6               ' Curve to the left
CurveRight      CON     7               ' Curve to the right

' Servo pulse width rotations

FS_CCW          CON     850             ' Full speed counterclockwise
FS_CW           CON     650             ' Full speed clockwise
NO_ROT          CON     750             ' No rotation
LS_CCW          CON     770             ' Low speed counterclockwise
LS_CW           CON     730             ' Low speed clockwise
```

Arduino



- Hardware
  - microcontroller:
  - peripherals: motors, sensors(rangefinder, ultrasonic, IR, PIR, sound, humidity, gas, gyro), servos, speakers, GPS, SD card adapters, lights, displays (including touchscreen), readers (RFID, smartcard)
- Software: C/C++

Arduino Sketch

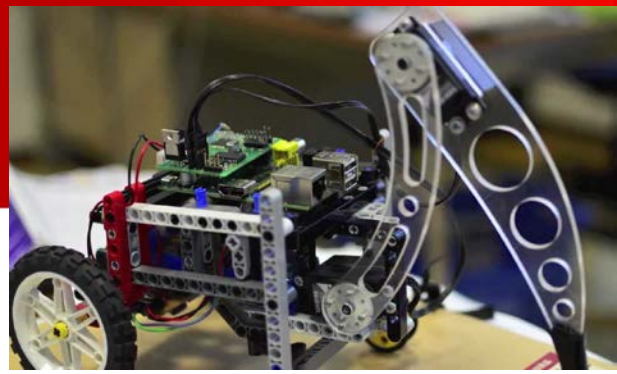
R02_Line_Follow | Arduino 1.6.5

File Edit Sketch Tools Help

R02_Line_Follow

```
/* Robot Line Follow

 This sketch demonstrates the line following capabilities
 of the Arduino Robot. On the floor, place some black
 electrical tape along the path you wish the robot to follow.
 To indicate a stopping point, place another piece of tape
 perpendicular to the path.

 Circuit:
 * Arduino Robot

 created 1 May 2013
 by X. Yang
 modified 12 May 2013
 by D. Cuartielles

 This example is in the public domain
 */

#include <ArduinoRobot.h> // include the robot library
#include <Wire.h>
#include <SPI.h>

long timerOrigin; // used for counting elapsed time

void setup() {
  // initialize the Robot, SD card, display, and speaker
  Robot.begin();
  Robot.beginTFT();
  Robot.beginSD();
  Robot.beginSpeaker();

  // show the logots on the TFT screen
  Robot.displayLogos();

  Robot.drawBMP("lf.bmp", 0, 0); // display background image

  Robot.playFile("chase.sqm");  // play a song from the SD card

  // add the instructions
```

Raspberry Pi 2



- Hardware
  - CPU: Arm Cortex A7, 32 bit, 1 GB memory, 40 I/O
  - Peripherals : Arduino shield, wifi, camera, additional device shield, display touchscreen
- Software: various linux distributions[raspbian], windows 10 IOT
- Additional Information: product is a full computer which has robotic applications

Presenters

John Mativo

jmativo@uga.edu

Roger Hill

rbhill@uga.edu

William Immonen

immonen@uga.edu

References

- Sumobot - http://www.parallax.com/microcontrollers/microcontrollers-overview
- Lego EV3 - http://www.ti.com/lit/ds/symlink/am1808.pdf
- general : http://www.futurlec.com/ICAtmel.shtml
- Raspberry Pi 2: https://www.element14.com/community/docs/DOC-73827/l/the-new-raspberry-pi-2-model-b-1gb-technical-specifications
- Robo Robo processor ATMEGA81: http://www.atmel.com/images/atmel-8159-8-bit-avr-microcontroller-atmega8a_datasheet.pdf
- Vex tiva ARM Cortex-M4: http://www.ti.com/lit/sg/spmt285d/spmt285d.pdf
- vex robot brain: http://www.vexrobotics.com/brain-g.html
- Arduino Robot: https://www.arduino.cc/en/Main/Robot